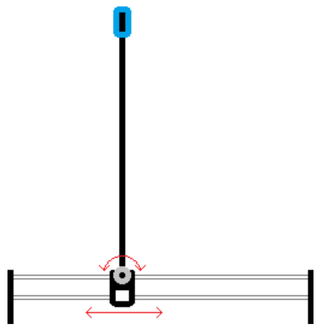
Inverted Pendulum

The accompaning Matlab code for this example is contained in Design_CartPend_4.1m and



Nominal State Space Model [umich]

Parameters

$M = 0.2$ [Kg] Cart Mass

$m = 0.2$ [Kg] Pendulum Mass

$l = 0.3$ [Meters] Length to pendulum center of mass

$b = 0.001$ [N/m/sec] Cart friction coefficient (10% uncertainty)

$I = 0.006$ [Kg.m^2] Mass moment of inertia of pendulum

$g = 9.8$ [m/s^2] Gravity

F: [Newtons] Force applied to cart

θ: [Radians] Pendulum angle from vertical (down)

x: [Meters] Cart position

Inverted Pendulum State Space Equations

$$m\frac{\partial^2 x}{\partial t^2} + ml\ddot{\theta}cos\theta - ml\theta^2 sin\dot{\theta} = N$$

$$I\frac{\partial^2 \theta}{\partial t^2} = -Plsin\theta - Nlcos\theta$$

$$ml\frac{\partial^2 \theta}{\partial t^2} = Psin\theta + Ncos\theta - mgsin\theta - m\ddot{x}cos\theta$$

$$M\frac{\partial^2 x}{\partial t^2} = F - N - b\frac{\partial x}{\partial t}$$

which can be rewritten as

$$(I + ml)\ddot{\theta} + mglsin\theta = -ml\ddot{x}cos\theta$$

$$(M + m)\ddot{x} + b\dot{x} + ml\ddot{\theta}cos\theta - ml\dot{\theta}^2 sin\theta = F$$

Nominal state space model

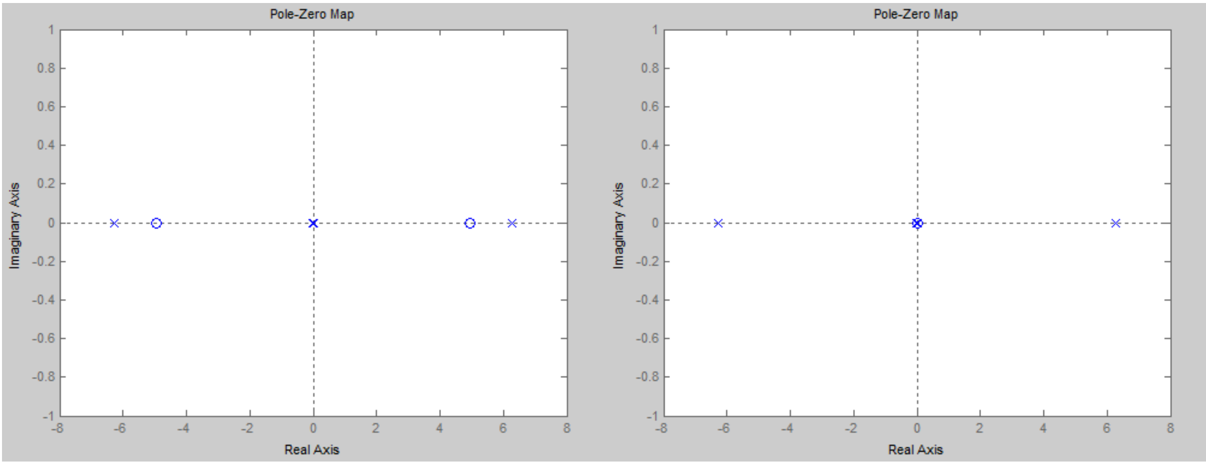
$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-(I+ml^2)*b}{I(M+m)Mml^2} & \frac{m^2gl^2}{I(M+m)Mml^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-mlb}{I(M+m)Mml^2} & \frac{mgl(M+m)}{I(M+m)Mml^2} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{I+ml^2}{I(M+m)Mml^2} \\ 0 \\ \frac{ml}{I(M+m)Mml^2} \end{bmatrix} u$$

$$\begin{bmatrix} x \\ \theta \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u$$

Pole/Zero Analysis

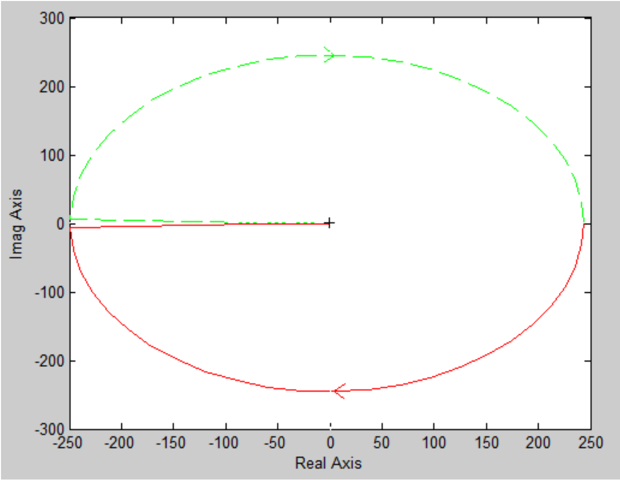
The plant has poles at [0, -0.0025, -6.2617, 6.2602], conjugate zeros at [$\pm$4.9497] in the first channel, and two zeros near the origin in the second channel at [1.0e-017*0.1737, 0] These were obtained using the pole command in Matlab.

Channel 1 left, channel 2 right



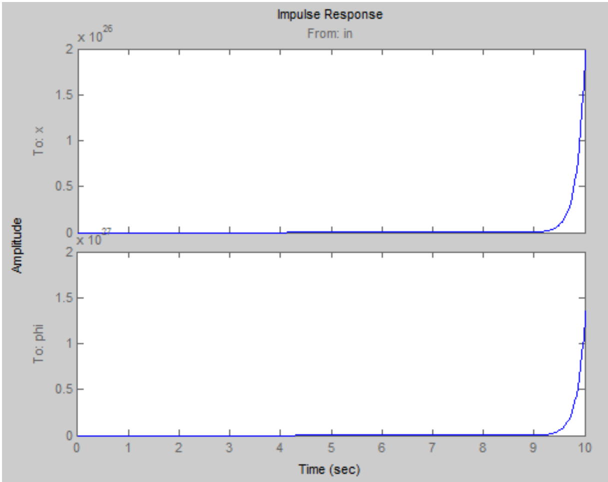
Nyquist

Instability is confirmed in the Nyquist plot as there are no encirclements of -1, while the open loop system has 1 pole in the RHP.



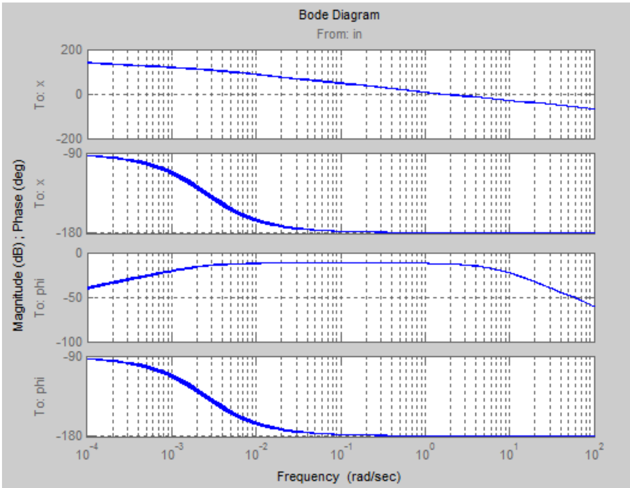
Impulse response

As expected from the pole/zero diagram the impulse response is unstable. Impulse was used here instead of step as it is more applicable to a perturbation about an equilibrium point.



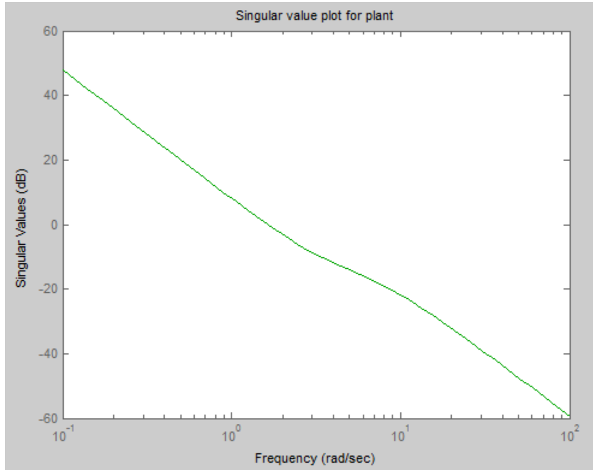
Bode

Here channel 1 corresponds to the cart input to the cart position output, and channel 2 corresponds to the same input but to the pendulum angle output. The difference between the two channels is clear, as the cart has more gain at lower frequencies and it's gain drops off with frequency, whereas the pendulum has very little gain across all frequencies, and what gain there is exists in a passband region. This lack of gain in the pendulum channel can probably be attributed to the plant being under-actuated, with the pendulum position being dependent on the cart, and the relative amount of force input required to restabilise the perturbed pendulum.



Singular values

The singular value plot below shows the effective gain across all channels and all frequencies, since the plant is SIMO.



Uncertainties

Uncertainties in control can be broadly categorised into two types, parametric and frequency weighted. A parametric uncertainty is the simplest to define the the most precise representation in most cases, but is a "real" uncertainty and hence the hardest to optimise for. A frequency weighted uncertainty uses a frequency domain weight to approximate the uncertainty caused by one or several uncertainties in a system, with the shape of the filter following the frequency domain effects of the uncertainty. The magnitude of the weight represents the magnitude of the uncertainty and the frequency ciuitoff reflects where the frequency where the uncertainties become less effective in the system. In practice the approximation can be quite crude yet still be effective. The frequency domain approximation is the easiest to solve for as it can just be posed as minimisation over frequency.

Uncertainties in the plant were limited to a 10% real parametric uncertainty in the cart friction coefficient to simplify the uncertain plant, as the size of the plant grows dramatically with the increase in the number of uncertainties. There is scope for inclusion of further real uncertainties, repeated real uncertainties, and for one or two complex uncertainties for the unmodelled high frequency dynamics of the pendulum and cart. The real uncertainties were modelled in Matlab using the ureal command and uss.

Control Objective

The inverted pendulum is quite a difficult control problem to solve as it contains three significant problems. It is non-linear, though it is locally linear in small perturbations about π and $-\pi$ respectively. It is unstable as a result of a pole in the open right half plane. It is under-actuated since it has one input to the cart, but two outputs namely the cart position and pendulum angle. It is however intuitively controllable, which can be proven using the controllability matrix. The control problem is primarily one of stabilisation of the pendulum, but with a secondary reference tracking objective in the cart position channel. In a realistic implementation it might also be necessary to limit the maximum deviation in the cart position due to physical constraints. It seems quite common in the literature to tackle the inverted pendulum problem with the LQR and LQG control methods, and LQR and pole placement methods are much quicker to get good results with.

Requirements summary:
- stable (all poles in RHP)
- track cart position as a secondary requirement
- minimise pendulum deviation from it's equilibrium
- moderate limits on control effort
- robustness to uncertainties in the plant
- care needs to be taken with the units and responses. Inputs are in meters, outputs are in radians, these units are huge.

S/KS Design [gu, skogestad,orinal paper]

The first technique to consider is the S/KS mixed sensitivity $\mathcal{H}_\infty$ minimisation problem. The basic principle of this method is to minimise in the sense of the infinitive norm, the transfer functions from the disturbance inputs to the performance outputs, under the assumption of a unit input. Mathematically this is expressed as

$$\|F_l(P,K)\|_\infty < \gamma$$

The value γ represents the upper bound on the infinitive norm of the system across its inputs and outputs, and γ_{min} represents the minimum achiveable γ value for the system. The optimal value using the Ricatti equation method is found by gamma iterations, whereby the Ricatti equations are solved for a given γ value, and then γ is reduced, repeating until the Ricatti equation has no solution. Although the solution at γ_{min} is optimal it is often numerically inconditioned and so a solution corresponding to a higher γ is usually chosen. The infinitive norm is used as it is the peak or worst case gain of the transfer functions, and hence can be used to give a maximum upper bound. . It's inverse of γ, is termed ϵ the stability margin, and can be used to show how much uncertainty can be tolerated whilst maintaining stability in the system.

A particular advantage of the $\mathcal{H}_\infty$synthesis procedure is that it always produces a stabilising controller, quite appropriate for the application of an unstable inverted pendulum. This is due to $\mathcal{H}_\infty$synthesis using Q parameterisation which expresses the set of all stabilising controllers as function of a free parameter Q!

The problem is formulated as an interconnected plant, usually denoted P, which relates the disturbance inputs to the performance outputs. It also relates the control inputs to the control outputs, but these don't form part of the minimisation problem as they are used to close the loop with the controller via LFT. The partitioning of the interconnected plant P is shown below. The (B_w, C_z) pair are the performance channel(s) and the pair (B_u, C_y) are the plant input/output channel(s).

$$P = \left[\begin{array}{c|cc} A & B_w & B_u \\ \hline C_z & D_{11} & D_{12} \\ C_y & D_{21} & D_{22} \end{array}\right]$$

In most cases the problem is subject to 5 main conditions:

1) $D_{22} = 0$

This requires that there be no direct link between the input and the plant, since it defeats the purpose of having a controller. Synthesis can still be conducted with this condition violated though the formulas are more complex.

2) (A, B_u)stabilisible and $C_y, A)$detectable

3) $D_{12} = \begin{bmatrix} 0 \\ I_{m2} \end{bmatrix} D_{21} = \begin{bmatrix} 0 & I_{p2} \end{bmatrix}$

4) $\begin{bmatrix} A - j\omega I & B_2 \\ C_1 & D_{12} \end{bmatrix}$ has full column rank $\forall\omega$

5) $\begin{bmatrix} A - j\omega I & B_1 \\ C_2 & D_{21} \end{bmatrix}$has full row rank $\forall\omega$

In the most common situation, where $D_{22} = 0$, a simplified controller formula can be obtained. The controller requires the solutions X, Y, of the following two Algebraic Ricatti Equations

$$A^T X + XA - XBB^T X + (1 - \gamma^{-2})^{-1} C^T C = 0$$

$$AY + YA - YC^T CY = 0$$

These correspond to two Hamiltonian matrices. The solutions X, Y have a spectral radius that is bounded by γ^2

$$\rho(X, Y) < \gamma^2$$

The controller is then obtained with the following formula

$$K_{opt} = \left[\begin{array}{c|c} A - BB^TX - (1-\gamma^{-2})^{-1}ZYC^TC & ZYC^T \\ \hline B^TX & 0 \end{array}\right]$$

$$Z = (I - \gamma^{-2}YX)^{-1}$$

The synthesis procedure alone produces a stabilising controller, but other requirements are usually defined for a system, and these are included through the use of weighting functions on the loops, and optimised by minimising the weighted loop over frequency. The most common synthesis configurations are S/KS and S/T/KS, but any combination of transfer functions can be minimised albeit the results might not be very good. The S/KS problem corresponds to minimising the sensitivity and controller gain transfer functions. Some examples of the transfer functions are the open loop function L, sensitivity function S, the complementary sensitivity function T, and the control function KS. Some of these can also be expressed at either the input or the output. They can be derived using a block diagram of the system and following the loop.

$$L = GK$$

$$S = (I + GK)^{-1} = (I + L)^{-1}$$

$$T = GK(I + GK)^{-1} = GKS = 1 + S$$

$$KS = K(I + GK)^{-1}$$

When we say a performance input we typically mean a noise input, reference input, or disturbance input. The performance measures should also have realistic physical meaning.

TODO

- S at high omega should be below the inverse of the S weight
- T at low omega should be above the T weight
- diag for this

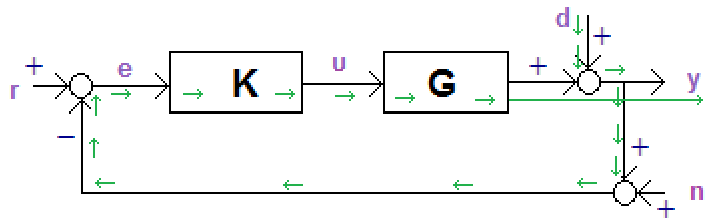
The S/KS Control Design for a Cart Pendulum

The S/KS design uses an objective plant with weights on control effort (Wu) and performance (Wpa). The control weight is a simple scalar. A more complex weight could be chosen to reflect the high frequency limitations of physical actuators where gain is likely to be approach zero, to prevent the controller from thrashing actuators near their frequency limits. Alternatively an actuator weight could be modelled “in loop” but this has been left out to simplify design. The performance weight has been chosen as a low pass filter to weight the cart position error highest at low frequencies. No weight was placed on the pendulum as the only requirement upon it is that it be stabilised and hence return to zero. A low pass weight was chosen because in S/KS synthesis the performance weight is a weight on the sensitivity function S, which is the transfer function from disturbance input to the plant output. This can be seen from the following equation for the $\mathcal{H}_\infty$ objective to be minimised

$$\left\| \begin{array}{c} W_pS \\ W_uKS \end{array} \right\|_\infty < \gamma$$

Ideally the closed loop would have good disturbance rejection up until the cutoff frequency of the plant and zero after, since the higher frequency dynamics are unmodelled. This is good illustration of how closed loop objectives can be described as constraints on the open loop model.

This diagram shows graphically how S relates d to y



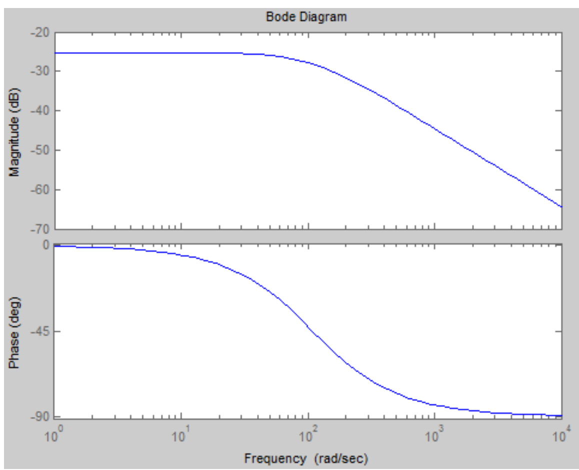
The control weight was chosen as

$$Wu = 20$$

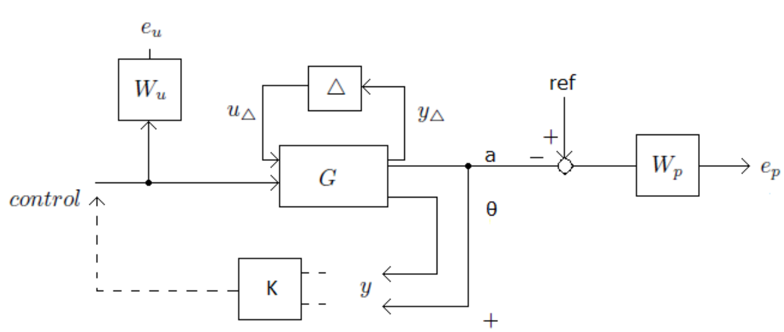
and the performance weight was chosen using makeweight as: 0.05* makeweight(1.1,50,0) resulting in:

$$Wp = \left[\begin{array}{c|c} -109.1 & 8 \\ \hline 15 & 0 \end{array}\right]$$

The frequency response of the performance weight is



The weighted plant block diagram is given below. This is used for synthesis of the plant.



Mixed sensitivity synthesis was used to find a controller, using the hinfsyn command. Here the inputs are the plant, the number of measurements and the number of controls.

[Khinf CL GAM] = hinfsyn(wplant1, 2, 1);

The synthesis procedure resulted in a final gamma value of 3.9954 and hence the controller stabilises the plant.

Frequency Domain Analysis

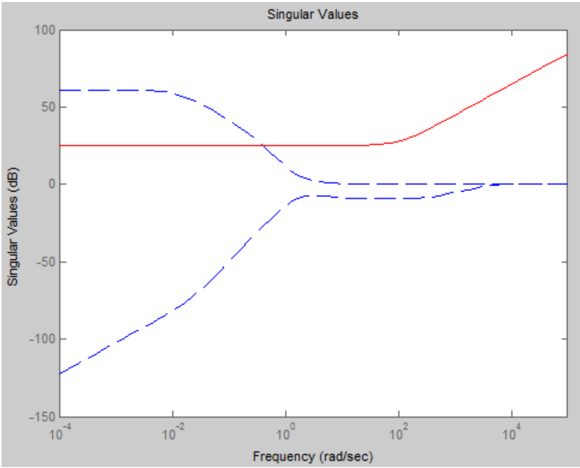
The singular values of the Sensitivity function are plotted with the singular values of the inverse $\mathcal{H}_\infty$ weight. Ideally the sensitivity function will be below the inverse $\mathcal{H}_\infty$ weight at low frequencies, which indicates that the plant has sufficient disturbance rejection at low frequencies. The reason for using the inverse $\mathcal{H}_\infty$ weight can be illustrated with the following equation

$$|W(j\omega)S(j\omega)| < 1 \quad \forall\omega$$

which can be rearranged as

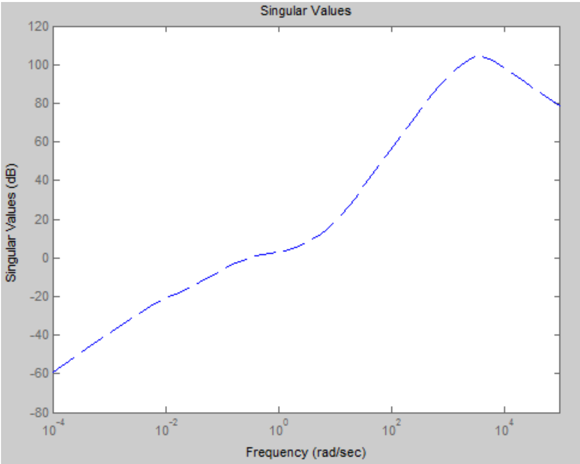
$$|S(j\omega)| < \frac{1}{W(j\omega)} \quad \forall\omega$$

This plot shows S vs the inverse performance weight in the first channel (Cart). At high frequencies the weight on S needs to be largest to put large attenuation on high frequency disturbances. Low frequencies disturbances are ignored as pendulum arm disturbances are fast. This needs to be a weight on the pendulum arm deflection from vertical.

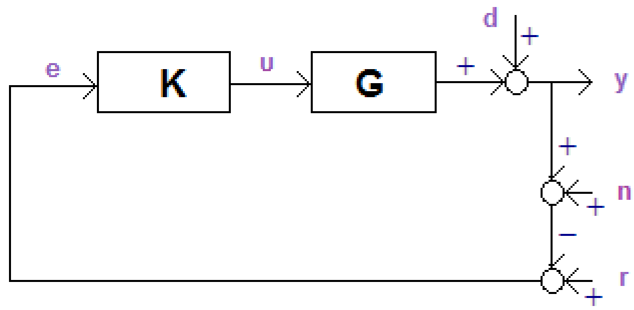


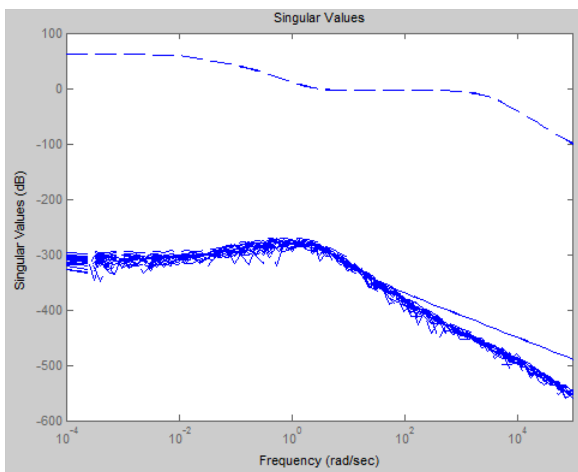
Weight on KS: Larger weight as frequencies increase to reflect actuator limitations. DC gain to reflect actuator power limitations.

The singular value plot of the controller and control effort weight is given below. If the singular values of K are below the inverse control weight then this indicates that the controller input limits have not been exceeded.



Although no weighting was defined on the complementary sensitivity function a plot of the singular values of T is still given below as T is indicative of reference tracking and noise attenuation performance. Ideally T would be large across all frequencies since large T at low frequencies indicates good reference tracking, and large T at high frequencies will cause noise to be attenuated as $y = T - noise$. However due to the fact that one cannot have both S and T large at a given frequency, the reference tracking is deemed the more important here. The situation with T is illustrated in the block diagram for T below, with the reference and noise inputs rearranged for clarity

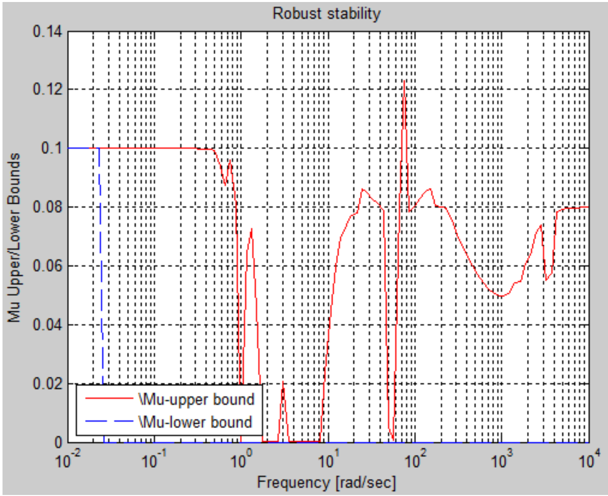




Mu robust stability analysis

The Mu analysis is presented in the graph below. For robust stability to be present the upper bound of mu should be less than 1 across all relevant frequencies, which in this case has been achieved as the upper Mu bound is approximately 0.1. The block structure was taken from the uncertain plant using lftdata. Evaluating the 3rd output (block structure) from the lftdata command confirms that the uncertain system has 1 real uncertainty with 2 occurances, and hence structured uncertainty analysis is required.

TODO: some theory about mu analysis



The analysis was performed by extracting the uncertainty block structure using lftdata and then closing the upper loop using an LFT with the uncertainty. The lower loop was then closed by creating a system with sysic by interconnecting the uncertain model with the controller, relating the reference and disturbance inputs to the two performance weight and control weight outputs. It is possible to use a lower LFT to close the loop, but this method allows us to specify a plant from w->z at the same time. An uncertain frequency response was then caclulated on the system and the results passed to robuststab() to obtain the mu analysis. The data in info can be used to produce the mu chart above.

The bounds are obtained directly from the command as:

UpperBound: 10.0000 LowerBound: 8.1328 DestabilizingFrequency: 0.0100

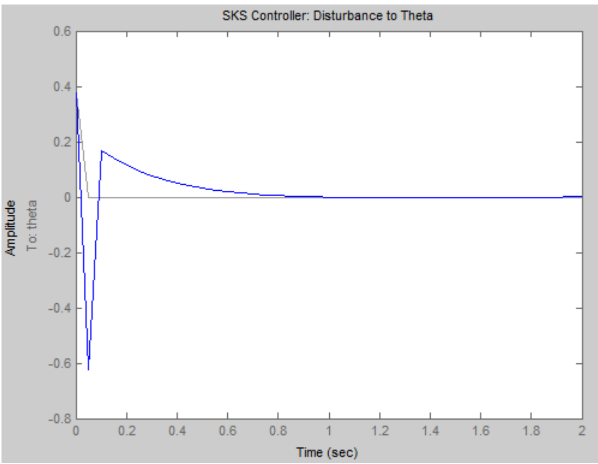
The lower mu bound indicates that the uncertain system is robust to 810% of the modelled uncertainty and becomes unstable at 0..0100 rad/sec.

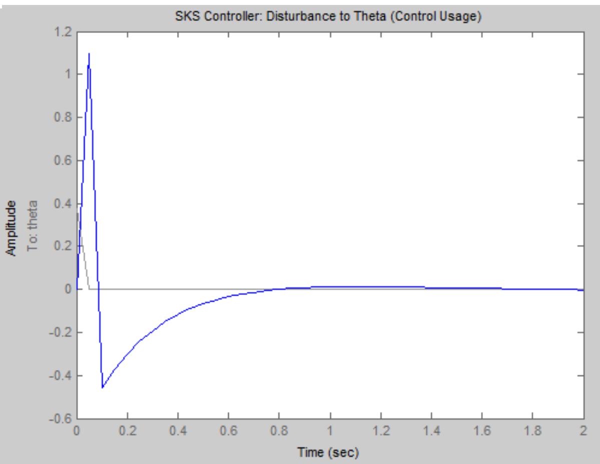
Time Responses

Then time responses were produced using sysic system interconnections from the disturbance to the pendulum angle and the reference to the cart position as these are the most relevant channels for control. Time responses are also shown of the control energy for these same two channels. The disturbance was modelled with a 0.375 impulse input for 1/40th of a second, and the reference was modelled as a unit step.

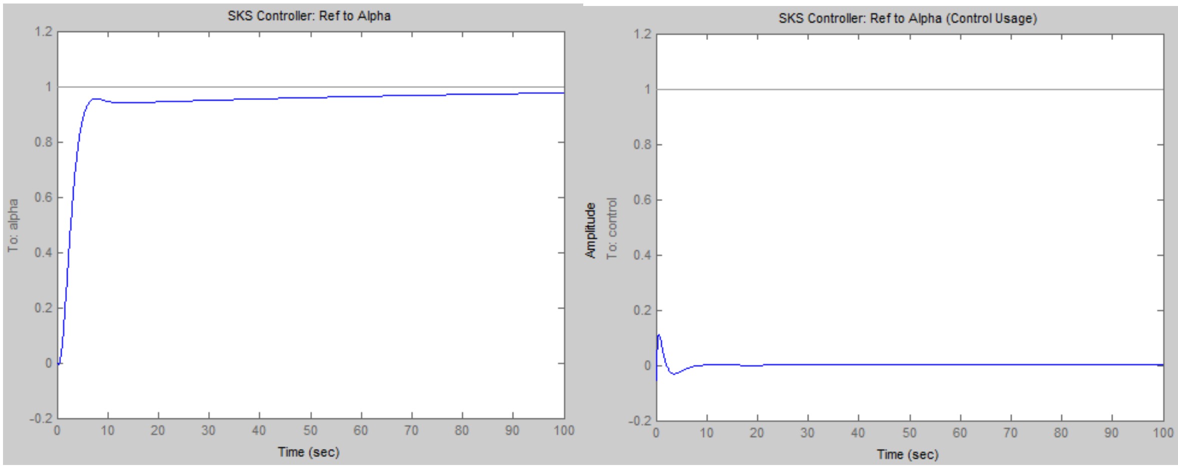
TODO: made this perturbed responses!

Disturbance Rejection





Reference Tracking



Mu Synthesis [skogestad, gu,
Structured uncertainty [Nagamune]

When considering uncertainty for an LTI system with some kind of structure it is necessary to generalise the singular value σ to the structured singular value μ. This is necessary as designs for plants with unstructured uncertainty may be conservative if used for those plants with structured uncertainty, as unstructured uncertainty approximates to the worst case. We can most easily see this in the SISO case. The inverse of μ is defined as the smallest structured perturbation $\triangle$, $\triangle \in \Delta$, such that the feedback system becomes unstable and so $det(I - M(j\omega)\triangle(j\omega)) = 0$. Assuming the uncertainty set is norm bounded such that $\|\triangle\|_\infty \leq 1$

$$\mu^{-1}_{\triangle(M(j\omega))} = \overline{\sigma}(\triangle(j\omega)) : det(I - M(j\omega)\triangle(j\omega)) = 0 \forall\omega$$

The maximum singular value of the perturbation at any given frequency which makes the perturbed system unstable, analysed for all perturbations and frequencies under considerion.

$$\mu_{\triangle(M(j\omega))} = \min_{\triangle(j\omega)\in\triangle} \frac{1}{\overline{\sigma}(\triangle(j\omega)) : det(I - M(j\omega)\triangle(j\omega)) = 0}, \forall\omega$$

As the uncertainty is structured, all admissable perturbations are assumed to belong to a given uncertainty set $\triangle$, and so have the structure that it defines. In the case that no uncertainty in the specified set causes the feedback system to become unstable then μ is defined as 0, and in the case that μ is 1 or greater then the feedback system is unstable.

The requirement for μ to be strictly less than 1 comes from the small gain theorem, where the norm bounded pertubations must be less than equal to 1, and the norm bounded system interconnection strictly less than 1. The determinant stability condition, $det(I - M(j\omega)\triangle(j\omega)) = 0$ can be considered as an extension of the SISO Nyquist stability criteria to MIMO systems.

The value of μ is used as a measure of how much uncertainty can be tolerated in the system before the system becomes unstable. The frequency at which the value of μ is greatest is the frequency at which the system is closest to instability. The value of μ at that frequency can also be used to determine what percentage increase in uncertainty of that within the set can be tolerated before instability is reached. In the normalised case mu should be less than 1, in the unnormalised case it should be less than beta where $\|\triangle\|_\infty \leq \frac{1}{\beta}$

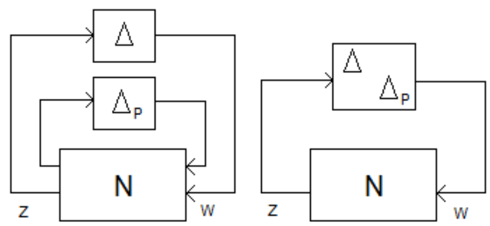
Mu synthesis is a method for obtaining controllers that have robust stability and robust performance, an improvement upon the $\mathcal{H}_\infty$synthesis procedure which only produces controllers that are robustly stable. The key behind this is to define the performance specifications in terms of uncertainty, a so called fictitious uncertainty, with the robustness evaluated using the stucture singular value (SSV). There are several synthesis procedures for obtaining a controller with RSRP (robust stability, robust performance), probably the most common being DK iteration. Mathematically this takes the following form

$$\hat{\triangle} = \begin{bmatrix} \triangle & 0 \\ 0 & \triangle_P \end{bmatrix} \quad [skogestad]$$

Where

$$\triangle = diag\{\delta_1 I_{r_1}, ...\delta_s I_{r_s}\} : \delta_i \in \mathbb{C} and \triangle_P = diag(\triangle_1, ..., \triangle_f) : \triangle_j \in \mathbb{C}^{m_j x m_j} \quad [Gu]$$

$$\sum_{i=1}^{s} r_i + \sum_{j=1}^{f} m_j = n$$



The two figures show how the main loop theorem is used to combine the parametric and performance uncertainties in to one augmented uncertainty block.

Robust performance requires a nominally stable unperturbed system $\|M\|_\infty < 1$or a robustly unstable plant may be synthesised. A robust performance problem can be posed in two ways, either as an analysis problem or as a synthesis problem. A structured uncertainty problem may be addressed in the same way, but with performance criteria excluded. The N structure is said to have robust performance if $F_u(P, \triangle) < 1$ for all norm bounded structured perturbations in the set.

Summary of mu conditions [Gu]
NS=> N internally stable
NP => $\overline{\sigma}(N_{22}) = \mu_{\triangle_P} < 1, \forall\omega$ and NS
RS=>$\mu_\triangle(N_{11}) < 1, \forall\omega$and NS
RP=> $\mu_{\hat{\triangle}}(N) < 1, \forall\omega, \hat{\triangle} = \begin{bmatrix} \triangle & \\ & \triangle_P \end{bmatrix}$ and NS

<u>DK Iteration</u>
To solve the robust stability robust performance problem, the objective is to find the controller K that minimises the upper bound of the SSV of the generalised plant for the frequency range of interest.

$$\inf_{K(s)} \sup_{\omega \in \mathbb{R}} \mu_\triangle(M(j\omega)) \quad [Gu]$$

The mu synthesis problem can be solved iteratively using the D-K or mu-K synthesis methods. There is no easy way obtain the upper mu bound analytically so an iteritive procedure is used to try to find it. A scaling matrix D is introduced such that $\mu_\triangle(M)$ is replaced by it's upper bound $\overline{\sigma}(DMD^{-1})$, using the scaling property that $\overline{\sigma}(M) = \overline{\sigma}(DMD^{-1})$. This follows since we are seeking the suprenum over omega and:

$$\rho(M) \leq \mu(M) \leq \overline{\sigma}(M)$$

A solution is then found by modifying D and K sequentially whilst holding the other stable and minimising $\left\|DMD^{-1}\right\|_\infty$. The resulting controller will have high order, equal to the number of states in the plant, weights and twice the number of states in D combined, so model order reduction techniques are often implemented.

A good solution in practice is to choose a suboptimal controller with gamma 5% above gamma min which results in a mix of a $\mathcal{H}_2$ and $\mathcal{H}_\infty$ optimisation. It is best to keep the uncertainty fixed and only vary the performance specifications.

The DK iteration procedure follows these steps [Gu]:

1. K step. Choose an initial real rational stable scaling matrix D, such as the identity matrix. Solve a standard H_∞ optimisation problem to obtain an initial stable controller K.

$$K = arg \inf_K \left\| F_l(\overline{P}, K) \right\|_\infty \quad [2]$$

2. Determine the mu curve for the controller K over a suitable frequency range and then normalise.

$$\mu_0(j\omega) = \mu(F_l(P, K)(j\omega)) \quad [2]$$

$$\bar{\mu_0} = \frac{\mu_0}{\max_\omega \mu_0} \quad [2]$$

3. D-Step. Scale $D(j\omega)$ to minimise $\overline{\sigma}(DND^{-1}(jw))$ for the frequency range of interest. Approximate each element of the scaling matrix with a low order stable, minimum phase transfer function $D(s)$.

4. Subsequent K steps. Obtain new controller

$$K_1 = \arg \inf_K \left\| \mu_0 F_l(\bar{P}, K) \right\|_\infty \quad [2]$$

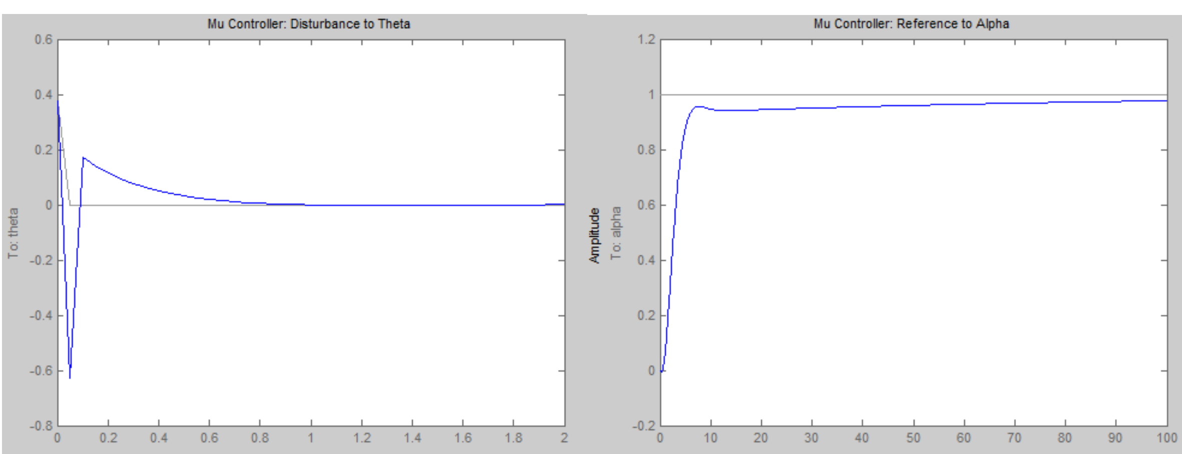
In subsequent iterations multiply the new mu curve on to the previous curves

$$K_2 = \arg \inf_K \left\| \mu_1 \mu_0 F_l(\bar{P}, K) \right\|_\infty \quad [2]$$

<u>Procedure and results</u>

Mu synthesis was used to obtain a controller using DK iterations and the same plant above, with a resulting mu value of 5.0094. The synthesis procedure used was the same as for the mixed sensitivity synthesis, but used the dksyn command instead. The objective here was simply to produce a robustly stable controller such that the DK iteration and SKS synthesis methods could be compared.

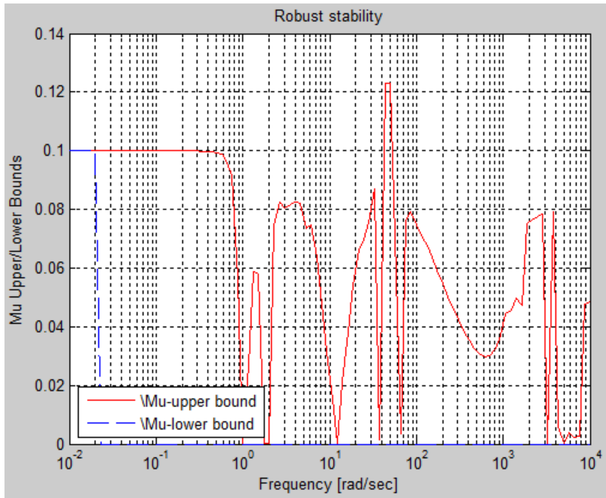
Time Responses



Robust Stability Analysis

The lower mu bound indicates that the uncertain system is robust to 812% of the modelled uncertainty, essentially the same as the mixed sensitivity controller.

UpperBound: 10, LowerBound: 8.1250 DestabilizingFrequency: 0.01Hz



Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ Synthesis with Pole Placement Constraints

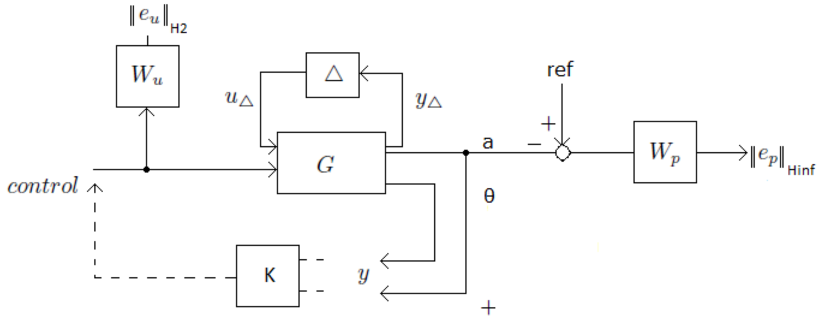
The third control method chosen was$\mathcal{H}_2/\mathcal{H}_\infty$ mixed synthesis without complex pole placement constraints which is a combination of the $\mathcal{H}_2$ optimal control problem and $\mathcal{H}_\infty$ robust control problem. The main improvement in this synthesis method is the capability to optimise subject to both energy optimality and peak robustness constraints. The synthesis method uses Linear Matrix Inequalities in the internal code with pole placement constraints, and even if no region is specified the "default" region is the open left half plane. After a baselane controller was established more advanced pole placement constraints were introduced. The constraints can be defined either by hand or using the automated LMIREG command. The work in this section builds on a previous built I did so I had a fairly good idea of what I wanted from the outset.

The control problem can be described in state space form as [Gahinet]. The first equation is the LTI state equation of the plant, the second and third are the output equations for the $\mathcal{H}_\infty$ and $\mathcal{H}_2$ objectives respectively, and the fourth is the output of the plant. Note particularly the equation for the $\mathcal{H}_\infty$ optimisation which has an i subscript due to Matlab inferring the number of $\mathcal{H}_\infty$ objective outputs from the number of outputs and the number of $\mathcal{H}_2$ objectives which is given.

$$\dot{x} = Ax + B_1 w + B_2 u$$

$$\begin{aligned} z_\infty &= C_i x + D_{i1} w + D_{i2} u \\ z_2 &= C_2 x + D_{21} w + D_{22} u \\ y &= C_y x + D_{y1} w + D_{y2} u \end{aligned}$$

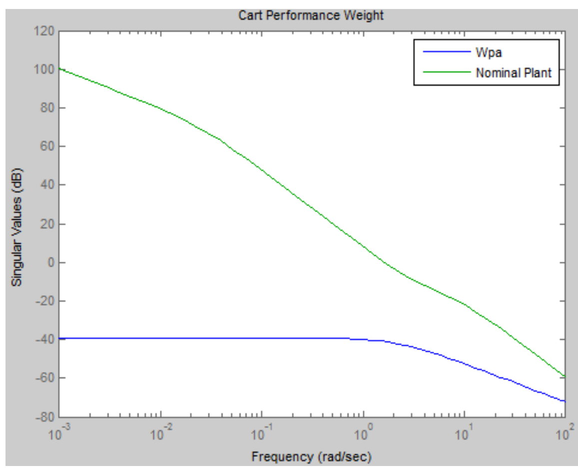
The control objectives are defined using three weights: A control effort weight, an $\mathcal{H}_2$ optimal performance weight and an $\mathcal{H}_\infty$ robustness weight. The weights were chosen to be the same as the S/KS mixed synthesis controller to make comparisons easier. More aggressively tuned weights are likely possible as all I've done is put them in the zero crossing region with some crossover band separation and tweak the gains a bit. The synthesis plant takes the same form as before but the control usage weight is minimised using the 2-norm whereas the performance weight is minimised using the infinitive norm.



Performance weight

The performance weight is of the same form to previously but has been adjusted to have a lower gain and a lower cut off frequency. It was created with makeweight as 0.01*makeweight(1.1,1,0).

$$Wpa = 0.01 \cdot \begin{bmatrix} -2.182 & 2 \\ 1.2 & 0 \end{bmatrix}$$



Control weight (Same as before)

The synthesis procedure was set up for $\mathcal{H}_2$ minimisation subject to $\mathcal{H}_\infty < 20$ to achieve a balance between reference tracking, control effort, and robustness. Default pole placement constraints were used, requiring poles to be placed in the open left half plane. This was done by calling the hinfmix function with the following arguments. The nominal plant was used as the system had to be converted to the old robust control form as hinfmix errors with version 3.

$$hinfmix(P1, r, obj)$$

The vector r contains the number of h2 objectives, the number of measurements and the number of controls

$$r = \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

The vector obj defines the optimisation objectives in terms of $\mathcal{H}_2$, $\mathcal{H}_\infty$, and tradeoff values between them. The values chosen correspond to h2 minimisation subject to $\mathcal{H}_\infty < 20$. In this case I chose to scrafice some $\mathcal{H}_\infty$ optimality in the performance channel for some $\mathcal{H}_2$ optimality in the control channel.

$$obj = \begin{bmatrix} 20 & 0 & 0 & 1 \end{bmatrix}$$

The hinfmix command is an interface for specifying an LMI control problem without having to specify the entire LMI problem manually in code which greatly improve usability.

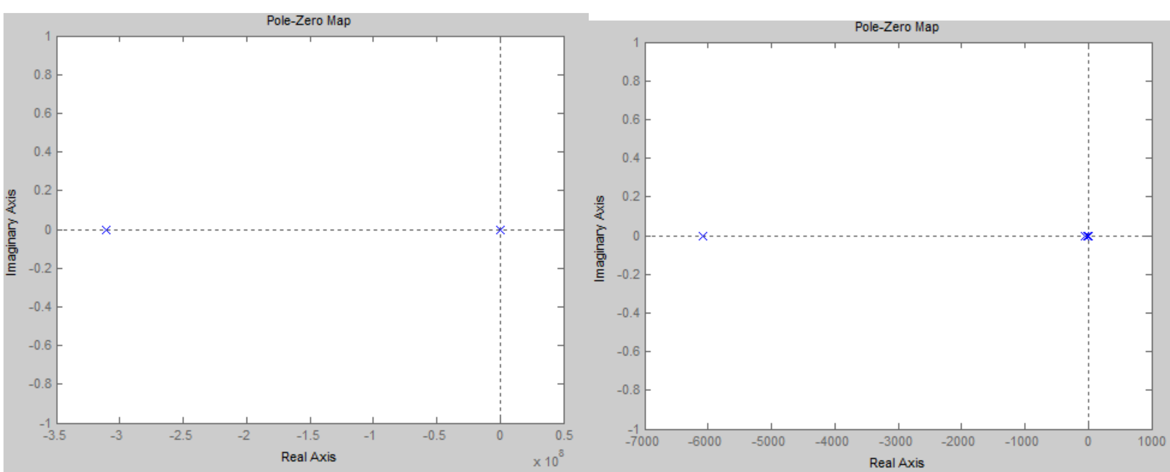
The optimal $\mathcal{H}_2$ and $\mathcal{H}_\infty$ values achieved were:
Guaranteed $\mathcal{H}_\infty$ performance: 20 (This was specified)
Guaranteed $\mathcal{H}_2$ performance: 0.0129
This shows how powerful the mixed synthesis method is as a it allows great control over the tradeoff between the two. In loopshaping the control over this tradeoff is limited to just reducing the optimality of the $\mathcal{H}_\infty$ solution.

Pole Placement Constraints
With the baselane controller established we can now cover the development of a controller with region constraints. The synthesis procedure is essentially the same as the above, so I shall focus on the pole placement. The constraints considered here are half planes, discs and cones. Half planes are used for placing poles further into the left half plane, increasing the real part of the poles, leading to a faster response time. Discs are used to constrain the undamped natural frequency of a poles. Cones are used enforce minimum decay rates, with greater angles from the origin having higher levels of damping.

The configuration first chosen was a disc region centered at the origin with a radius of 10000. This clearly includes the right half plane, but as the synthesis procedure produces stable controllers the effective region is the left half of the disc. The purpose of this constraint is illustrated by the pole zero diagram of the controller without pole constraints, where very high frequency poles are present, which is hardly a desirable situation. The radius of 10000 constrains undamped poles to at most 10000 radians/second, moving the high frequency poles.

$$region = 1.0e+004 \cdot \begin{bmatrix} -1.0000 + j0.0002 & 0 & 0 & 0 \\ 0 & -1.0000 & 0.0001 & 0 \end{bmatrix}$$

Pole/Zero Diagram without pole placement constraints and with pole placement constraints



Pole/Zero Diagram without pole placement constraints
1.0e+008 *
-3.1109 -0.0000 -0.0000 0.0000 -0.0000

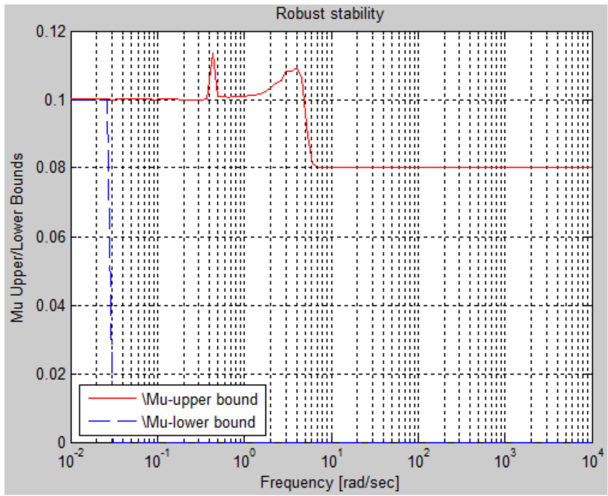
Pole/Zero Diagram with pole placement constraints
1.0e+003 *
-6.0860 -0.0520 -0.0233 0.0011 -0.0080

With the pole placement constraint synthesis the optimal $\mathcal{H}_2$ and $\mathcal{H}_\infty$ values achieved were essentially identical to those without pole constraints:

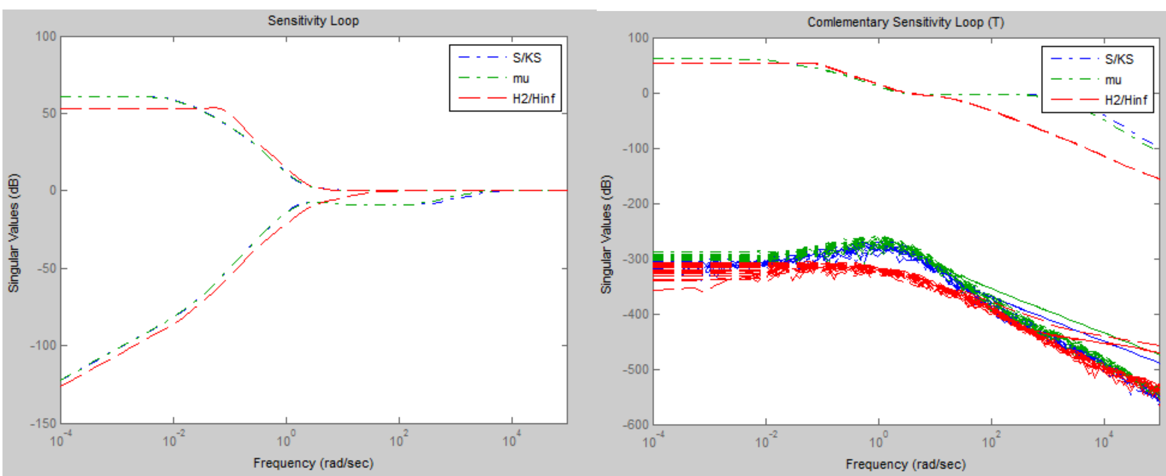
Guaranteed $\mathcal{H}_\infty$ performance: 20 (This was specified)

Guaranteed $\mathcal{H}_2$ performance: 0.0128

Mu Analysis

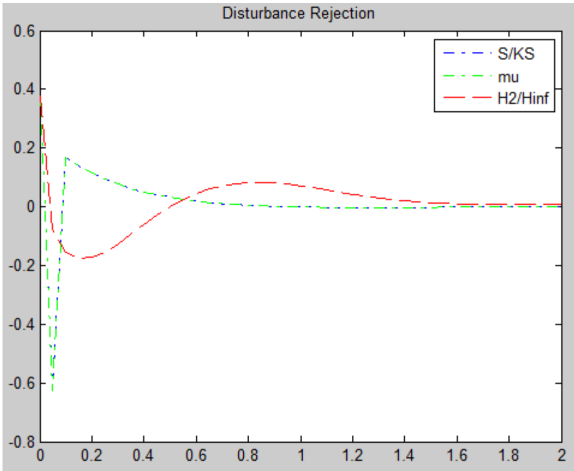


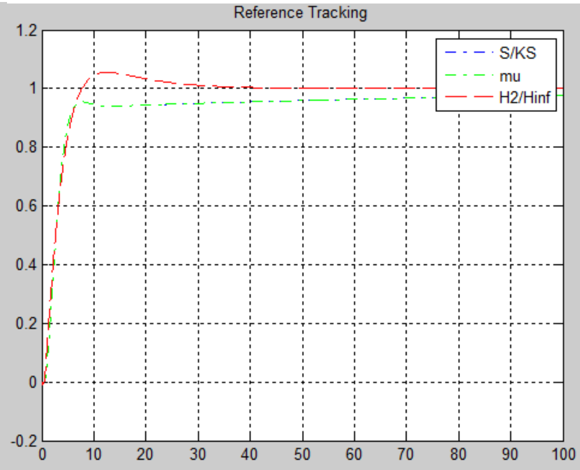
Frequency Domain Analysis



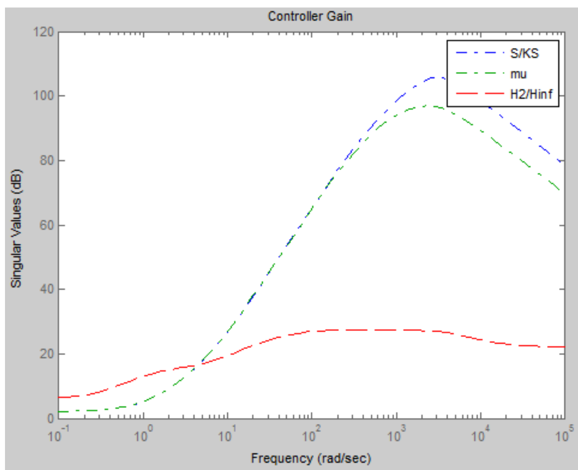
Closed Loop Analysis

The disturbance rejection and reference tracking performance of the mixed sensitivity, mu-synthesis and $\mathcal{H}_2/\mathcal{H}_\infty$mixed synthesis plants are shown in the diagrams below. From the reference tracking diagram we can see that the controllers all perform similarly, with the $\mathcal{H}_2/\mathcal{H}_\infty$controller having some overshoot but zero steady state error. From the disturbance rejection diagram we can see that the mu and s/ks controllers have a faster disturbance rejection but have sharp changes in enegy usage, whereas the response of the $\mathcal{H}_2/\mathcal{H}_\infty$controller is more realistic having a gradual, albeit oscillatory disturbance response. The $\mathcal{H}_2/\mathcal{H}_\infty$ controller does also have the advantage of better tuned weights.





When evaluating the controller gain comparison diagram the differences are very substantial, with the sharp changes in the mu and s/ks disturbance responses likely to be due to the much larger high frequency gains in those controllers.



Conclusion

This control problem was by far the most difficult with the most problems to overcome, both due complexities in the plant and problems with the synthesis software. The differences in the responses of each channel make it quite hard to find weights which gave good results for both, and trying to tune both channels at the same time is particularly hard as the plant is under-actuated. The cart channel is a quite normal system with gain reducing with frequency, whereas the pendulum has a passband with not much control gain in it. The MIMO nature of the system means singular values were used for design instead of Nyquist/Root Locus/Bode, but these plots are given anyway as they can be interesting. Matlab presents almost endless issues, far too many to list, which are solved with combinations of the documentation and experimentation. The performance of both the LSDP and hinfinity synthesis controllers is quite reasonable. The main aim of this problem was to demonstrate and experiment with a greater range of Matlab control features which I definitely think was accomplished.

References

[1] http://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum&section=ControlStateSpace

[2] Gahinet. Pascal., Nemirovski. Arkadi., Laub. Alan., and Chilali. Mahmoud. Matlab LMI Control Toolbox Users Guide.

Nagamune. R. (2009) Mech 550F: Multivariable Feedback Control Lectures, Department of Mechanical Engineering, University of British Columbia.

skogestad
gu
manc equations of pend

TODO
- new conclusion required!
- explain controller energy required. put in terms of volts rads, not just dB
- nyquist isn't analysed properly nor explained. open loop nyquist shows stability of closed loop system. if open loop has P open loop RHP poles then there must be -P counterclockwise encirclements of -1 for the system to be stable.
- bode isn't properly explained
- perturbed responses
- S</inv(wp) for nominal performance
- robust performance analysis (add a complex pert for this)
- nonlinear simulation (simulink)